

## **APPENDIX A**

*System Development Methodology*

**Table of Contents**

1	Introduction .....	2
2	System Development Life Cycle .....	3
2.1	Process Driven Design .....	3
2.2	Business Driven Design .....	3
2.3	System Development .....	4
2.4	Business Analysis .....	5
3	The Project Team .....	7
4	Project Phases .....	9
4.1	Business Analysis Phase .....	9
4.2	System Analysis Phase .....	11
4.3	Design Phase .....	14
4.4	Construction Phase .....	16
4.5	Testing Phase .....	18
4.6	Implementation Phase .....	20
4.7	Production Phase .....	22
4.8	Post Implementation Review Phase .....	23
5	Glossary .....	24

## 1 Introduction

This guide specifies the standards used to develop business application systems. It will serve as a reference to all project team members.

This document describes a staged approach to methodically engineer a business system by following a prescribed life-cycle. It follows an information engineering concept which emphasizes data-driven system design instead of process-oriented system design. Joint Application Design (JAD), a design session where Management, Users, and Parties involved meet with Information Technology professionals to jointly design an Information system, is utilized here to define the user requirements.

In our experience with implementing business information systems, we have found that those which yield the highest business value are those which support an Information Architecture derived from business functions. To achieve these high business value system implementations, Database Engineering uses an Information Engineering methodology to identify business functions and their related information model.

"Information modeling is a technique for describing information structures. Logical models specify business information requirements, policies and rules. Physical models specify solutions. As such, information modeling can be viewed as a way to develop information system requirements and solutions. Models are used to design a physical system database that represents the specific needs of a particular implementation project and the general needs of the business area in which that project is being conducted."<sup>1</sup>

The Information Engineering approach to information requirements analysis has been extensively researched and documented by our industry's critical thinkers. James Martin, John Zachman, Robert G. Brown and Thomas A Bruce, Richard Barker, and other information engineering methodologists have published specification languages that help us understand and describe information structures. Recently Object Oriented extensions have been added to Information Engineering to help us model business events and data driven rules.

We use methodology as a mechanism for:

- Managing risks;
- Producing predictable results;

---

<sup>1</sup> Bruce, Thomas A. *Designing Quality Databases with IDEF1X Information Models*. New York, NY.: Dorset House Publishing, 1992.

- Facilitating communication among members of the project team.

The information engineering method is a "top-down" approach to information requirements analysis -- mission and objectives lead to business functions that in turn lead to information architecture. System requirements are derived from information architecture. It is an approach that results in an implementation independent statement of an organization's business requirement for information.

## 2                   **System Development Life Cycle**

### .1                   **Process Driven Design**

Traditional system design is process driven, analyst oriented, and sequential. The focus is on how the system is to look and behave, and the analyst is responsible for designing the system. The phases of a project are clearly defined and completed sequentially; the team moves to the next phase with the expectation that revisiting previous phases will only be necessary if "mistakes" have been made.

Diagram here

In process driven design, an activity or process (1) defines system function (2) which in turn defines data requirements (3).

Process driven design:

- Mechanizes a process
- Frequently has process (application) specific data stores
- Achieves integration by using a post-processing data management application
- Is used where process is stable
- Is used where process is critical

### .2                   **Business Driven Design**

This methodology follows an information engineering concept which is iterative, data-driven, and user oriented. In an information engineering methodology, the design focus is on "data" rather than "process", "what" the system has to do rather than "how" it has to do it. The users are full participants at all stages of the system development. The methodology allows for and expects that the team may move among the phases, as refining the team's understanding of the requirements is an iterative process.

Diagram here

In business driven design, business function (1) implies data architecture (2), which is implemented by an information system (3), which is supported by business activity (4), which executes a business function (1) and so on.

Business driven design recognizes that:

- business functions are more stable than business activities
- data architecture (data and the relationships among them) exists as a result of business functions and is independent of information systems
- information systems are the vehicle used to implement data architecture

### .3

### System Development Process

The system development process has eight phases. The phases are loosely defined as prototyping causes significant overlap.

- Business Analysis - understand the business
- Systems Analysis - define functionality required of a system to support the business
- Design - determine how to implement the Systems Analysis
- Construction - build and document the system
- Test - validate the system against the design (The Quality Assurance process)
- Implement - prepare for the production environment
- Production - optimize and move to production
- Post Implementation Review - what we learnt

Two techniques, Joint Application Design (JAD) sessions and Prototyping, allow this process to become iterative and user oriented. Both of these techniques encourage the users to provide input at all phases of development.

JAD sessions employ a team of Management, User, and Development professionals to jointly determine the user requirements. This normally consists of defining the user's business functions, developing a data model from the business functions, and then determining which portions of the data model are to be implemented in the system.

Prototyping is a strategy for building a system by starting with a working model based on the user requirements and then refining the model through an

iterative process. The objectives of prototyping are to ensure that the users and the system developers have a mutual understanding of the requirements and to then develop a finished product that meets the requirements.

The data model and the functional hierarchy of the system are prototyped during the Systems Analysis and Design phases. The look and feel of the system and the implementation of the business rules are prototyped during the System Analysis, Design, and Construction phases.

#### **.4 Business Analysis**

Business Analysis consists of:

- Function analysis
- Entity analysis
- Relationship analysis
- Data analysis

Function analysis starts at the top of the company hierarchy, with the development of vision and mission statements. Starting at the top helps us to focus on business function rather than process. We then work down through the corporate hierarchy developing mission statements until we reach the business group that we are performing the business analysis for. We then use one simple (verb/noun) sentence to describe the whole business function. The business is then broken down into 6-12 major functions. Each of these functions is broken down further until the team reaches the lowest level that is appropriate. No reference is made to mechanism (the way the work is done now) because the mechanism may change. Each function must be examined to see if it supports the mission statement. This helps to detect processes that no longer have a valid business reason for existing.

Entity analysis involves reviewing the business functions and extracting things of significance about which information needs to be known or held. Usually, these will be the nouns in the business function sentences. The names assigned to "entities" should be consistent and meaningful to the user. For example, Invoice and Invoice Line Item.

Relationship analysis defines how entities are related to each other. They must be named in the context of the business functions. For example, Invoice may be made up of one or more Invoice Line Items. Invoice Line Items must be for one and only one Invoice.

Data analysis involves reviewing the entity analysis output in the light of recorded or known data values that can be studied (reports, files, charts). This is known as bottom up checking.

The Business Analysis document is then prepared using the information extracted from the analysis. The project team then begins preparing the Project Definition document. At part of this, they determine which

portions of the data model would benefit from automation. This is usually done by cost/benefit analysis.

If a decision is made to proceed, the project definition document is created. It contains the project objective, background, scope, and plan (methodology, team, development environment, and estimate). Appendices contain the business analysis, and functional specification. The business analysis document contains the vision and mission statements, the business functions and the data model developed from the functions. (This information is independent of the project and its scope.) The functional specification document contains information on the user interface, data maintenance functions, reporting facilities, system operations and administration functions, and interfaces to other systems.

All of these documents will be refined as the data model prototyping continues in the Systems Analysis and Systems Design phases.

## 3                   **The Project Team**

The project team will consist of the following groups and individuals.

### **The Project Sponsor**

It is the responsibility of the Project Sponsor (Client) to provide corporate commitment, make prompt decisions regarding the system design and implementations, resolve management issues, set priorities and provide formal acceptance of the system at various phases of the System Life Cycle.

### **The Management Representative**

It is the responsibility of the Management Representative (Client) to act as the liaison between the project team and user groups, to facilitate the resolution of management related issues and to oversee the development and implementation of management/administrative processes, policies and procedures.

### **The User Group**

It is the responsibility of the User Group (Client) to participate in all phases of the project, define business functional requirements, confirm system analysis and design, participate in prototyping sessions, perform acceptance tests and take ownership of the completed system. Representatives of the user groups will confirm completion of the project phases by signing the appropriate documents. The users will take ownership of the completed application.

### **Project Manager**

It is the responsibility of the Project Manager (Client or consultant) to establish, coordinate, execute, monitor and report progress of all activities leading to a successful completion of the project. This includes making design decisions, planning and scheduling, the day-to-day management of the project, and chairing all sessions requiring user participation. The Project Manager will issue a monthly status report to the project team and the project executives indicating problems and issues that need to be discussed.

### **Project Development Team**

It is the responsibility of the Development Project Team (Client or consultant) to carry out assigned tasks within the given time frame. It is the responsibility of the Data Architect (DA) to participate in and review the business and system functional analysis phases, ensure logical integrity of the data model, conformance to the data architecture standards, conformance to procedures and standards.

## 4 Project Phases

### .1 Business Analysis Phase

#### Objectives

The objectives of the business analysis phase are to:

- Understand and describe the user's business functions and problems
- Determine whether an automated system is required
- Prepare a project definition
- Start preparing a functional specification document

#### Tasks

Review request for project and:

- Establish project and system environment
  - Investigate business objective
    - Develop business function hierarchy
    - Document existing policies
  - Review business objective with management
  - Establish data usage pattern
    - Develop Entity/Relationship model
- Prepare project definition
  - Describe project objective, background, scope and plan
  - Define project team
  - Prepare functional specification
    - Describe at functional level the user interface, data maintenance functions, reporting, system operation and administration, and interfaces
- Prepare detailed plan for the System Analysis phase
- Review project plan and report on progress

#### Events

- Project start-up
- Business Analysis walkthrough (including Data Model)

- Business Analysis sign-off
- Project definition walkthrough
- Project definition management review
- Project definition sign-off
- Data model prototyping cycles

**Input**

- Project Specific

**Deliverables**

- Business Analysis Document including Business Model
- Project Definition including Functional Specification
- Project Authorization
- Detailed project plan for the System Analysis phase

**.2****System Analysis Phase****Objectives**

The objectives of the system analysis phase are to:

- Complete Functional Specification document
- Produce a business case for the project
- Determine and document the detailed business requirements of the system

**Tasks**

- Review Business model and Project Definition
- Develop business case
  - Complete Functional Specification
  - Do cost/benefit analysis
- Prepare detailed Data Models
  - Define entities (key, attributes, volumes)
  - Define relationships
  - Update detailed Entity/Relationship Diagrams
- Specify Elementary functions
  - Define function logic
  - Define function frequencies
  - Define the response mode required
- Establish cross-check matrices
  - Develop function/entity matrix
  - Develop function/organization unit matrix
  - Develop entity/organization unit matrix
- Define system operation requirements
  - Define backup/recovery requirements
  - Define data retention requirements
  - Define audit/security requirements
  - Define distributed requirements
- Define data conversion requirements
- Prepare detailed plan for the Design phase
- Review Project plan and report on progress

### **Events**

- Decision to proceed with project
- Joint Application Design - Detailed Business Functions
- Joint Application Design - Detailed Data Modelling
- Structured Walkthroughs
- Conversion requirements review
- System Analysis Document sign-off

### **Input**

- Project Definition
- Business Analysis Document
- Detailed project plan for the System Analysis phase
- Existing application file layouts

### **Deliverables**

- Functional Specification
- System Analysis Document
- Detailed project plan for the Design phase

**.3****Design Phase****Objectives**

The objective of the design phase is to:

- Determine how to implement the recommended system

**Tasks**

- Perform logical/physical data design
  - Produce table/view design from data model
  - Perform necessary de-normalization
  - Define indexes, clusters, sizing estimates, space allocation
- Develop the logical application design
  - Map functions onto modules
  - Develop module specifications
  - Specify design standards for screens, dialogues, menus, and security (database level, table level, and application level)
  - Specify auditing requirements
  - Specify data backup requirements
  - Design procedures, pre-printed material, paper forms
- Develop the physical application design
  - Choose the implementation tools
  - Acquire/install the required software, hardware, and network
  - Design guidelines for the tools
  - Start prototyping
- Prepare System Testing plan
- Set up Development and Test environment
- Review specifications and obtain approval
  - Review database design
  - Review application design
- Develop data conversion specification and procedure
- Prepare detailed plan for the Construction phase
- Review Project plan and report on progress

**Events**

- **System Design walkthrough**
- **System Design Document sign-off**
- **Prototyping cycles**

### **Input**

- **Functional Specification**
- **System Analysis Document**
- **Detailed project plan for the Design phase**

### **Deliverables**

- **System Design Document**
- **Hardware/Software purchase order/installation**
- **System test plan**
- **Development and Test environment**
- **Detailed project plan for the Construction phase**
- **Prototypes**

**Objectives**

The objectives of the construction phase are to:

- Code, test and integrate all system components as defined in the Design phase
- Complete User Manual, Operation Manual, education material and System Technical Document

**Tasks**

- Review System Design document
- Produce programs
  - Prototype
  - Code and unit test each program
  - Perform integration testing
  - Revise code as necessary
  - Revise program documentation
- Finalize System Design Document
- Develop documentation
  - Develop User Manual
  - Develop user training plan and material
  - Develop System Technical Document
  - Develop Operation Manual
- Review system with users
- Prepare detailed plan for the Testing and Implementation phase
  - Define responsibilities
  - Define approach
  - Define test cases
- Review Project plan and report on progress

**Events**

- Sign-off final System Design Document
- Progress meeting
- Technical review (code walkthrough)
- Prototyping cycles

### **Input**

- System Design Document
- System Test plan
- Detailed project plan for the Construction phase
- Prototypes

### **Deliverables**

- System Design Document
- Detailed System Test plan
- Detailed Implementation plan
- Working, tested modules
- User Manual
- Operation Manual
- System Technical Document
- Training Plan
- Detailed project plan for the Testing/Implementation phase

## **.5 Testing Phase**

### **Objectives**

The objective of the construction phase is to:

- Confirm that the computer based system has met its design Objectives

### **Tasks**

- Prepare Acceptance Plan
- Migrate application to test environment
- Perform system test
- Evaluate test results and produce change request list
- Revise System Technical Manual, Operation Manual, and User Manual
- Review Implementation plan
- Review Project plan and report on progress

### **Events**

- System audit
- Technical review (code walkthrough)

### **Input**

- System Analysis Document
- System Design Document
- Detailed System Test plan
- Detailed Implementation plan
- Working, tested modules
- User Manual, Operation Manual, System Technical Document
- Detailed project plan for the Testing phase

### **Deliverables**

- System Test cases/results
- Acceptance test plan/cases

- System Technical Document
- User Manual
- Operation Manual
- Detailed Implementation plan
- Change request list
- 'Tested' application

**.6****Implementation Phase****Objectives**

The objectives of the implementation phase are to:

- Perform the acceptance test in the Test environment and prepare the Production environment
- Simulate the Production environment

**Tasks**

- Prepare and perform acceptance testing
  - Set up user accesses
  - Review test cases
  - Run test
- Evaluate acceptance test results and produce change request list
- Perform user training
  - Review training plan/material
  - Conduct training
- Set up Production environment
- Migrate application to production environment
- Perform data conversion
- Finalize User Manual, Operation Manual, and System Technical Document
- Obtain User Sign-off on the system
- Review Project plan and report on progress

**Events**

- Users training
- Test results review
- Users sign-off

**Input**

- Acceptance test plan/cases
- Detailed Implementation plan

- User Manual
- Operation Manual
- Training Plan
- System Technical Document
- Data conversion procedure
- 'Tested' application system

### **Deliverables**

- Training Plan/material
- Acceptance test results
- Change request list
- User Manual
- Operation Manual
- System Technical Document
- Production environment
- 'Production' application system
- System sign-off

.7

## **Production Phase**

### **Objectives**

The objectives of the production phase are to:

- Optimise the computer-based system within the Production environment
- Manage the operation of the system

### **Tasks**

- Set up users access
- Perform application tuning
- Monitor/review performance
- Set up system support
- Log and resolve system problems

### **Input**

- User Manual
- Operation Manual
- System Technical Document
- 'Production' application system

### **Deliverables**

- Operation/Problem log
- Performance statistics

**Objectives**

The objectives of the post-implementation phase are to:

- Audit all aspects of the system performance and to provide recommendations for improvement
- Comment on the project experience, indicating effective technique and strategy employed, problems encountered, and experience gained

**Tasks**

- Review Performance statistics
- Review Problem log
  - Review recovery/backup problem
  - Review system failure frequency
- Prepare report

**Input**

- Problem log
- Performance statistics

**Deliverables**

- Post implementation report

## 5                   Glossary

This glossary contains a list of terms used in this document.

**Acceptance Test:** A test (or a series of tests) used to simulate the live operation of the entire system including manual procedures, forms, and processing schedules. This could be a parallel test, a stress test, or a benchmark. The results of these tests are compared against the documented acceptance criteria and decisions are made to either implement the business system or further improve its operation.

**Application Tuning:** The method of improving the throughput or the speed of the operation of an application by adjusting the design, program statements, and other parameters. Although based on careful measurement and observation of the system using performance monitors, application tuning depends heavily on trial and error, and 'feel'. In Oracle applications, application tuning involves optimising the SQL statements, indexing, and adjusting the table design.

**Attribute:** Any detail that serves to qualify, identify, classify, quantify or express the state of an entity; or any description of 'a thing of significance'. Attributes are the simplest, smallest pieces of information about entities and are literal values, such as an integer, character string, date, etc.

**Audit Requirements:** The need to set up transaction logging so that every transaction can be traced. In Oracle, transaction logging can be set up by means of triggers and dedicated tables, or Oracle's own auditing facilities can be used.

**Backup/Recovery:** Backup is a procedure for creating a copy of the current application data for safekeeping. Recovery is a procedure for recreating the application data from the backup copy. We suggest that all Oracle application data have daily, weekly, and monthly EXPORT, as well as daily operation system backup. If an application requires addition backup, the designer should specify the crucial data, the appropriate backup method, the backup cycle, time of backup, and off-site storage requirements.

**Business Analysis Document:** A document that describes the results of the business analysis conducted. This document should contain the following:

- Mission of the business
- Function Hierarchy
- Entity/Relationship Diagrams
- Description of the Relationships (optionality, cardinality)

**Business Case:** A document that describes the benefits to the business of performing some action (in our case, developing an application). Benefits

could include: profitability, staff performance, availability of management information, morale, etc. Includes cost/benefit analysis as well as more qualitative benefits.

**Business Function:** What a business does or needs to do, irrespective of how it does it.

**Business Model:** A representation of the processes involved in the conduct of business activities and the major categories of information employed by those processes. To describe your organization's requirement for information, *you use names, terms and expressions that have specific or unique meaning to your organization*. This consists of Function Hierarchy, Entity and Relationships, and the Entity/Relationship Diagrams. This set of business vocabulary, together with its definition, is called the business model. The various pieces of the model are called the components or business objects.

**Business Objectives:** A statement of business intent that may be measurable or quantifiable; for example, to increase profit by 1% during the next quarter.

**Business Term:** A Business Term is a word or expression that conveys a business meaning when it is applied to an entity. Although Business Terms and Attributes have similar uses, Business Terms are differentiated by how they will be implemented. They are either a qualifier, a function or a list of values.

**Change Request List:** A list of outstanding items that do not meet requirements and need corrective actions.

**Cluster:** A concept in Oracle. It is an alternative storage arrangement in which data from more than one table may appear in the same block. The physical storage is arranged to optimise a given join operation.

**Company Strategic Plan:** A plan consisting of the objectives of the company, the resources used to obtain those objectives, and the policies that are to govern acquisition, use, and disposition of resources.

**Cost/Benefit Analysis:** A quantitative measure of the cost and benefits of providing a system. Used in Business Case documents.

**Data Conversion Requirements:** The requirements for loading the data from an old system to a new application. Points to consider are:

- Where is the data currently held?
- What is the format?
- What is the medium?
- What is the volume?
- How much of the existing data needs to be loaded?
- Is there any new data needed to be created?
- What are the security/privacy issues in handling the data?

**Data Model:** A representation of the entities identified in the Business Model, as well as the relationship between those entities.

**Data Modelling:** An analysis of the Business Model to identify entities, entity relationships, attributes, data usage and dependencies.

**Data Retention:** The period of time that the data should be retained on the current database before archiving.

**De-normalization:** The process of introducing some redundancy into the table for performance reasons. Such a process must be well documented and the appropriate update techniques written in to the application software.

**Dialogues:** A way of asking for information by reports and programs, and accepting response from the users through a computer terminal.

**Distributed Requirements:** Distributed requirements can be separated into Distributed Processing and Distributed Database. Distributed Processing allows an application to run on one node (computer) while accessing data stored on another node. The application accesses data on only one node, but many applications or copies of the applications can run on the same or different nodes. The processing is distributed but the data is located at one (central) node. A Distributed Database allows many physical databases, located at different nodes, but they appear to be a single 'logical database'. Different tables may be at different nodes. Tables may be 'partitioned' or 'fragmented' either horizontally or vertically.

**Domain:** A Domain is a business focus or a major activity of an organization. For example an organization's activities and interests related to the people it employs is usually referred to as the "Human Resources Domain".

**Elementary Business Function:** A business function which, if started, must be completed. It cannot exist in an intermediate stage. Elementary business functions are at the lowest level of a function hierarchy and cannot be further broken down.

**Entity:** An Entity is an object of significance in a Domain (a distinguishable person, place, thing, concept or event of interest to the company) about which information needs to be known or held.

**Entity/Organization Unit Matrix:** A matrix that shows which entities are referenced by each applicable organization unit.

**Entity Relationship (ER) Diagram:** A diagram depicting the association or connection between two or more entities.

**File Layout:** A description of the record structure stored in the file. This includes the field names, the data type and the field length.

**Function:** A Function is a value derived from one or more attributes. Age is an example of a function. It is information about an entity (a person, a building, a car etc.) derived from the difference between an original date (the birth date) and the current date. Defining a function is very similar to adding a column to a spreadsheet that is derived from two or more existing columns in the spreadsheet.

**Function Frequencies:** The frequency that a function is performed. This affects design of the module that implements this function.

**Function Logic:** Function logic is used to define the actions carried out by an Elementary business function. It is required when its meaning is not clear from the definition or from the Data Model. The function logic statement is a detailed description of the function naming the entities, attributes and relationships involved, qualifying them where necessary, stating when and how they are affected, and including any complex rules for calculating values. Structured English or Pseudo code can be used to specify the logic.

**Function/Entity Matrix:** A matrix used to indicate which functions are referred to by the entities and attributes on the data model. Entities have a function to: create occurrences, update occurrences, delete occurrences, read, archive or all.

**Function/Organization Unit Matrix:** A matrix used to identify which functions are executed by the applicable organization unit.

**Function Hierarchy:** A simple grouping of functions in a strict hierarchy, representing all the functions in an area of a business. This forms part of the Business Model.

**Index:** A means of accessing one or more rows in an Oracle table with particular performance characteristics. An index may quote one or more columns and be a means of enforcing uniqueness on their values.

**Information Resource Plan:** A plan consisting of the current environment definition, the data entity evaluation, processes, data usage, new application developments and interfaces.

**Integration Test:** The purpose of Integration testing is to check that all individual modules of the system provide the functionality, performance and reliability required of them by the System Design Specification.

**Joint Application Design (JAD):** A design session where management and users meet with the Information Technology professionals to jointly design an Information system through discussions and presentations.

**List of Values:** A List of Values is a set of data corresponding to a query specification. The list is derived from a query specification.

**Menu:** A common way of providing structured access to groups of modules.

**Module:** A program or procedure within an application, which usually performs one Elementary Business Function. (Several functions may be performed by a single module occasionally).

**Normalization:** A step by step process that produces either entity or table definitions that have:

- No repeating groups
- The same kind of values assigned to attributes or columns
- A distinct name
- Distinct and uniquely identifiable rows.

**Operation Manual:** A manual that provides a guide to day-to-day operations of a system. This should provide internal details of the system such that diagnosis of problems can be carried out by the operations staff when the problem occurs.

**Organization Unit:** An administrative or functional unit in an organization.

**Performance Statistics:** Statistics used to show how well a computer system performs as a whole, and how well its components interact with each other.

**Post Implementation Report:** A report produced during the Post Implementation phase to analyse the benefits, costs, system performance, and the project performance of a computer application system. Recommendations are also included to suggest future direction of the application, major system problem discussion and recommended solution.

**Problem Log:** A log to record the date, time, description, and parties involved in a system problem.

**'Production' Application System:** The collection of system directories (Source, Command, Executable, etc.) and databases residing on the Production environment.

**Program:** A set of computer instructions, which can enter, change or query database items, and provide many useful computer functions.

**Project Authorization:** A form used to authorize the start of a project. This form must have the signatures of the Executives who sponsor the projects, the project manager, and the appropriate users.

**Project Definition:** A detailed description of a project. It contains the following components:

- Project Objective - should be clear, achievable, and measurable.
- Project Background - a description of the problems or opportunity.
- Project Scope - identifies the boundary, major focal points, and the exclusions.

- **Interfaces** - identifies the system level impact as well as the organizational level impact.
- **Project Constraints** - identifies the limitations that the project has to work with.
- **Approach** - identifies the methodology used, project change control, status reporting and the acceptance criteria.
- **Project Structure** - identifies the project team members, the user groups, the Executives sponsoring the project, the responsibilities and the project approval authorities.
- **A high level project plan** - contains a high level task description in each project phase, the estimate and the schedule.

**Property:** A Property is any detail that serves to qualify, identify, classify, quantify or express the state of an entity. Entities have three types of properties: Attributes, Business Terms and Relationships.

**Qualifier:** A Qualifier is a label or word that differentiates instances of an Entity. Frequently, qualifiers are the words or phrases that are the choices available in look-up lists for attributes such as "type, status, etc.". "Active", "Full-time", "Unionised" are examples of qualifiers of the Human Resource Entity.

**Relationship:** A Relationship is a named, significant association between two entities. The relationship is the most complex property of an entity. It represents interdependencies among complex objects (entities). It is the relationship property that enables navigation among related entities.

**Response Mode:** The mode of response of an Elementary business function; I for Immediate, O for Overnight and translates to module response mode of 'interactive' and 'batch' respectively.

**Security:** The protection of the database against unauthorized use. In Oracle, there are four levels of security - database level, table level, function level and product level. The database levels are CONNECT, RESOURCE, or DBA options. The table levels are SELECT, UPDATE, INSERT, DELETE, ALTER, INDEX, and ALL. The function level is whether the user can execute certain functions supported by the application. The product level is whether the user can use a certain Oracle product or not.

**Screen:** A terminal display of a layout of text and data designed by the developer.

**Sizing Estimates:** A sizing estimate is made to determine the disc space requirements for a table or an index in Oracle.

**Space Allocation:** The process of assigning a storage location and a Space Definition to an Oracle table.

**Space Definition:** A Space Definition is a named set of storage parameters, and may be used by any Oracle table at the time that the table is created, by referencing the Space Definition in the CREATE TABLE statement.

**Structured Walkthrough:** A walkthrough is a review of a product. The walkthrough is said to be structured if it follows a formal set of rules.

**Subject:** A Subject is an area of specialization. "Staffing", "performance management", "benefits administration" are examples of subjects in the Human Resources Domain (see Domain).

**System Analysis Document:** A document that describes the results of the system analysis conducted. This document should contain the following:

- Function/Entity matrix
- Function/Organization Unit matrix
- Entity/Organization Unit matrix
- Function logic
- Entity volumes and function frequencies
- Distributed requirements
- Backup/recovery requirements
- Data retention requirements
- Audit/security requirements
- Data conversion requirements
- Subsequent plans

**System Audit:** A process to assess the system with respect to the following aspects:

- The implemented model versus the business model
- Naming convention, and standards
- Documentation completeness

**System Design Document:** A document that describes the results of the system design conducted. This document should contain the following:

- Logical and physical models, including
  - table/column design
  - view design
  - index design
  - program/module design
  - clerical procedure
- Detailed sizing estimates
- System test plan
- Specifications for module, audit, backup
- Access and security rules
- Data conversion specifications
- Subsequent plans

**System Technical Document:** A document that describes the physical components of the system and the relationships amongst them. This document should contain the following:

- Table definition (column, index)

- Column definition
- View definition
- Report description
- Module definition
- Module/table (view)
- Report/table (view)
- Table/view
- Menu structure

**System Test** : A System test is used to verify that the system components accurately perform the required business and control functions.

**Table**: A tabular view of data, which may be used on a relational database management system to hold one or more columns of data. It is often an implementation of an entity. Tables are the logical and perceived data structure, not the physical data structure, in a relational system.

**'Tested' Application System**: The collection of system directories (Source, Command, Executable, etc.) and databases residing on the Test environment.

**Training Plan**: A plan to identify the users to be trained, scope of training, training method, required sessions (scheduled, locations, contents and materials), trainers, and required facilities.

**User Manual**: A document that provides the function descriptions, screen/report/form/layouts, validation tables, error messages, and any general information appropriate to understanding the system.

**View**: A means of accessing a subset of the database as if it were a table. The view may:

- be restricted to named columns
- be restricted to specific rows
- change column names
- derive new columns
- give access to combination of related tables or view.